IN THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF NEW YORK

FIFTH GENERATION COMPUTER CORPORATION, Plaintiff, - against - INTERNATIONAL BUSINESS MACHINES CORPORATION, Defendant.	Case No.: 1:09-cv-2439 (JSR) ECF CASE

IBM'S OPENING *MARKMAN* BRIEF

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases

Unless otherwise noted, all emphasis is added.

I. Introduction And Background

On October 29, 2008, Fifth Generation Computer Corporation ("FGC") sued IBM, alleging that IBM's Blue Gene supercomputer infringed three patents: U.S. Patent No. 4,860,201 ("the '201 Patent"), U.S. Patent No. 4,843,540 ("the '540 Patent") ("the Stolfo Patents"), and U.S. Patent No. 6,000,024 ("the '024 Patent").[1] (Ex. E, FGC Complaint at ¶ 26.) FGC alleges that by 2003 it knew that IBM was infringing, but waited to file suit until the end of 2008. (*Id*. at ¶ 21.) In the interim, in 2006, the Stolfo Patents expired. Recently, FGC dropped its '540 Patent infringement claim, leaving only the expired '201 Patent and the '024 Patent in the case.

Both the '201 and '024 Patents claim that they improve existing parallel processing computer systems, where large numbers of processors are connected in a binary tree configuration. Although binary tree computer systems were known before the '201 Patent, the inventors sought to improve communications in those systems by developing a specialized input/output (I/O) device with dedicated circuitry to move information up and down the binary tree. The '201 patent claims a binary tree of interconnected I/O devices and processors.

The '024 Patent claims to improve the binary tree of the '201 Patent, by moving communications away from the I/O devices and to a binary tree of bus controllers that manage information flow throughout the system. The bus controllers provide communications links among the parallel processors and between the processors and a host computer that controls the binary tree. The '024 Patent claims this binary tree of bus controllers, their associated processors, and their connection to the host computer.

On May 19, 2009, FGC disclosed that it is asserting claims 1, 4, 7 and 8 of the '201 Patent, and claims 1-10 of the '024 Patent. The parties then exchanged initial lists of terms for

---

[1] The '201 Patent is attached as Exhibit B, the '024 Patent is attached as Exhibit C, and the '540 Patent is attached as Exhibit D.

construction, and on June 12, the parties exchanged proposed claim constructions. The parties have since conferred and agreed to a number of constructions of previously disputed terms. The parties' agreed constructions and a list of the disputed terms are attached as Exhibit A.

II. Legal Principles Of Claim Construction

Any patent infringement suit "necessitates a determination of what the words in the [patent] claim mean." *Markman v. Westview Instr., Inc.*, 517 U.S. 370, 374 (1996) (internal citations omitted). "'Claim construction' is the judicial statement of what is and is not covered by the technical terms and other words of the claims." *Netword, LLC v. Centraal Corp.*, 242 F.3d 1347, 1352 (Fed. Cir. 2001). "Ultimately the interpretation to be given to a term can only be determined and confirmed with a full understanding of what the inventors actually invented and intended to envelop with the claim. The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction." *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998) (internal citations omitted).

Patent claims generally are given their "ordinary and customary meaning," which is "the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (*en banc*). However, "the person of ordinary skill in the art is deemed to read the claim term … in the context of the entire patent, including the specification." *Id.*

A. Intrinsic Evidence

The intrinsic evidence (the claims, the specification, and the prosecution history) "is the most significant source of the legally operative meaning of disputed claim language." *SmithKline Beecham Corp. v. Apotex Corp.*, 403 F.3d 1331, 1338 (Fed. Cir. 2005). This is because it is "created at the time of patent prosecution for the purpose of explaining the patent's

scope and meaning," not for the purpose of litigation. *Phillips*, 415 F.3d at 1318.

1. Claims

It is a bedrock principle of patent law that claims of a patent define the invention to which the patentee is entitled the right to exclude. *Aro Mfg. Co. v. Convertible Top Replacement Co.*, 365 U.S. 336, 339 (1961). "[T]he claims themselves provide substantial guidance as to the meaning of particular claim terms." *Phillips*, 415 F.3d at 1314. "[T]he context of the surrounding words of the claim … must be considered in determining the ordinary and customary meaning of [a claim] term." *Wilson Sporting Goods Co. v. Hillerich & Bradsby Co.*, 442 F.3d 1322, 1328 (Fed. Cir. 2006).

2. Specification

The claims "must be read in view of the specification, of which they are a part." *Phillips*, 415 F.3d at 1315. "[T]he specification is always highly relevant to the claim construction analysis [and] is the single best guide to the meaning of a disputed term." *Id*. (internal citations omitted); *see also Netword*, 242 F.3d at 1352 ("The claims are directed to the invention that is described in the specification; they do not have meaning removed from the context from which they arose."). When the specification contains a definition given to a claim term, "the inventor's lexicography governs." *Phillips*, 415 F.3d at 1316.

3. Prosecution History

Claims "must be construed with reference to the file wrapper of prosecution history in the Patent Office." *Research Plastics, Inc. v. Fed. Packaging Corp.*, 421 F.3d 1290, 1296 (Fed. Cir. 2005) (quoting *Graham v. John Deere Co.*, 383 U.S. 1, 33 (1966)). "[T]he prosecution history provides evidence of how the [Patent Office] and the inventor understood the patent." *Phillips*, 415 F.3d at 1317.

A disclaimer of claim scope during prosecution can occur when an applicant amends or cancels claims in response to their rejection by the examiner, or when an applicant argues that a claimed invention is distinguishable over the prior art. *See Atofina v. Great Lakes Chem. Corp.*, 441 F.3d 991, 998 (Fed. Cir. 2006). Following a disclaimer, a court "cannot construe the claims to cover subject matter broader than that which the patentee itself regarded as comprising its inventions and represented to the PTO." *Microsoft Corp. v. Multi-Tech Sys., Inc.*, 357 F.3d 1340, 1349 (Fed. Cir. 2004).

4. Abstract

A patent's abstract, which is on the cover of the patent, is a "potentially helpful source of intrinsic evidence as to the meaning of claims." *Hill-Rom Co., Inc. v. Kinetic Concepts, Inc.*, 209 F.3d 1337, 1341 (Fed. Cir. 2000). For this reason, courts have "frequently looked to the abstract to determine the scope of the invention." *Id.*; *Tate Access Floors, Inc. v. Maxcess Techs., Inc.*, 222 F.3d 958, 966 n.2 (Fed. Cir. 2000).

B. Extrinsic Evidence

When construing claims, a court may consider certain "extrinsic evidence," including "expert and inventor testimony, dictionaries, and learned treatises." *Phillips*, 415 F.3d at 1317. But extrinsic evidence is viewed as "less reliable than the patent and its prosecution history in determining how to read claims terms." *Id*. at 1318. It is "less significant than the intrinsic record" and "unlikely to result in a reliable interpretation of patent claim scope unless considered in the context of intrinsic evidence." *Id*. at 1317-19. "[H]eavy reliance on the dictionary divorced from the intrinsic evidence risks transforming the meaning of the claim term to the artisan into the meaning of the term in the abstract, out of its particular context, which is the specification." *Id*. at 1321; *see also In re Johnston*, 435 F.3d 1381, 1384 (Fed. Cir. 2006).

III. U.S. Patent No. 4,860,201

The ’201 Patent describes a parallel data processing system that consists of a large number of processing elements arranged in a binary tree. (’201 Patent at 1:18-22.)

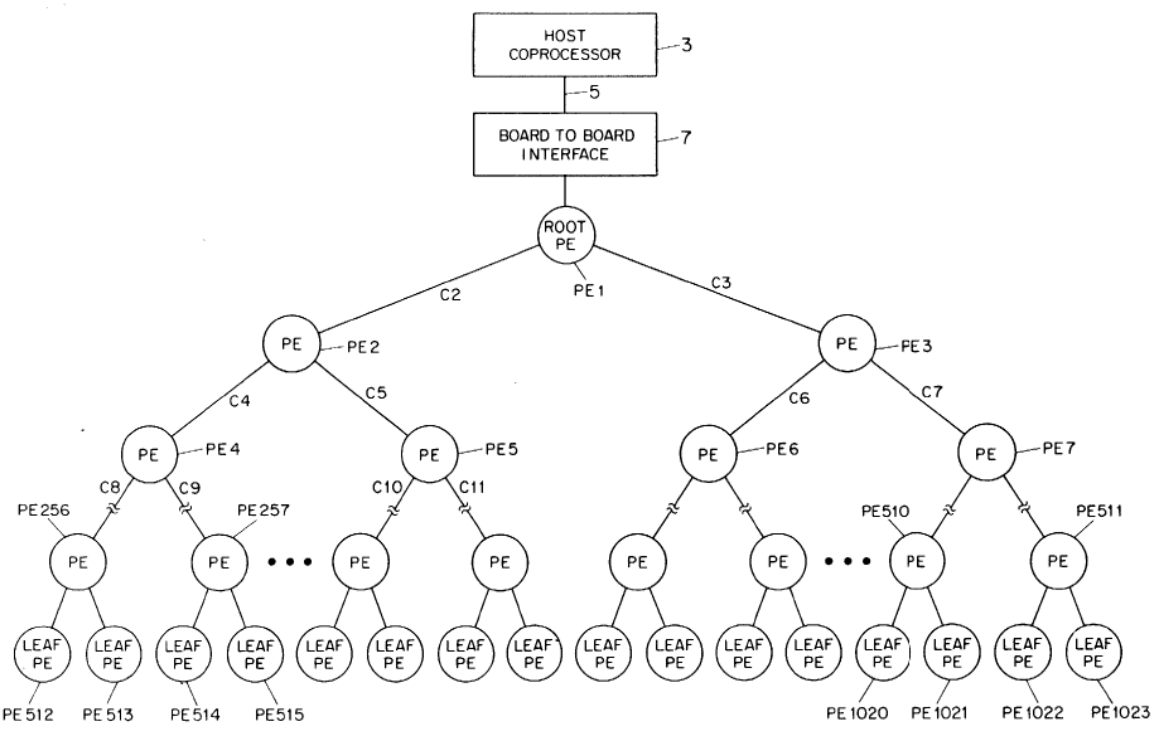

FIG. 2

As illustrated in Figure 2 of the ’201 Patent, “[i]n a binary tree computer, a large number of processors are connected so that each processor except those at the root and leaves of the tree has a single parent processor and two children processors.” (*Id*. at 1:62-65.) The processing element at the top of the tree is called a root processing element (Root PE) and is connected to a host computer instead of a parent processing element. (*Id*. at 8:47-52, 8:61-62.) Likewise, the processing elements at the bottom, or extremes, of the tree are called leaf processing elements and have no children. (*Id*. at 8:47-52.) The ’201 Patent applies this well-known tree structure to a special-purpose machine that is optimized to solve a narrow class of computational problems.

A. Stolfo Implemented A Binary Tree Structure To Speed Computation Of "Decomposable Searching Problems."

In the early 1980s, Dr. Stolfo and his colleagues at Columbia University sought to develop a computer system that could solve "decomposable searching problems." ('201 Patent at 4:29-39; Ex. F, '201 Prosecution History at FGC1277-79.)[2] A problem is "decomposable" if it can be solved by combining the solutions to arbitrary subsets of the problem. ('201 Patent at 2:41-44.) One such problem is the membership problem: "is x in F?" where F is a set of objects. (*Id*. at 2:14-18.) In a speech recognition application, F is a reference database of known words and x is a sample of speech to recognize. (*Id*. at 7:9-23.) This problem is decomposable, because the database of known words can be divided into subsets and the sample word "x" compared to each subset. If "x" is found in any subset, the word is recognized. (*Id*.)

Years before Stolfo began his work at Columbia, Carnegie Mellon University professors Bentley and Kung recognized that a binary tree of processors is well suited to solving a decomposable searching problem. (*Id*. at 1:43-61, 2:1-13, 2:45-56, 3:6-16.) Using a binary tree, F is first "decomposed" into subsets, so that each processor is assigned a particular subset, *i.e.* a set of known words. (*Id*. at 2:41-44.) Simultaneously, each processor looks for "x" — the speech sample — in its subset of known words, and reports its result up the tree to the root. (*Id*. at 2:48-56.) The processor that finds a match is the one that "recognizes" the speech sample.

According to Stolfo, prior art binary tree systems were not able to process this class of problems without "propagation delays" (*i.e.*, delays in the time it takes information to travel from one data processing element to another). (*Id*. at 3:44-47.) Delays were caused because those systems used processors both to perform the matching computations *and* to move information up

---

[2] The Stolfo and Shaw paper discusses ways to implement "production systems," a type of decomposable searching problem. ('201 Patent at 3:54-56.)

and down the tree (*Id*. at 3:17-33.) Thus, Stolfo searched for a way to speed up the binary tree implementation by focusing on processor-to-processor communications up and down the tree.

B. Stolfo Proposed A Specialized Input/Output Device To Speed Information Flow In the Binary Tree Computer System.

To speed data flow, Stolfo moved certain communications responsibilities away from the processors themselves and to specialized input/output (I/O) devices in each processing element.

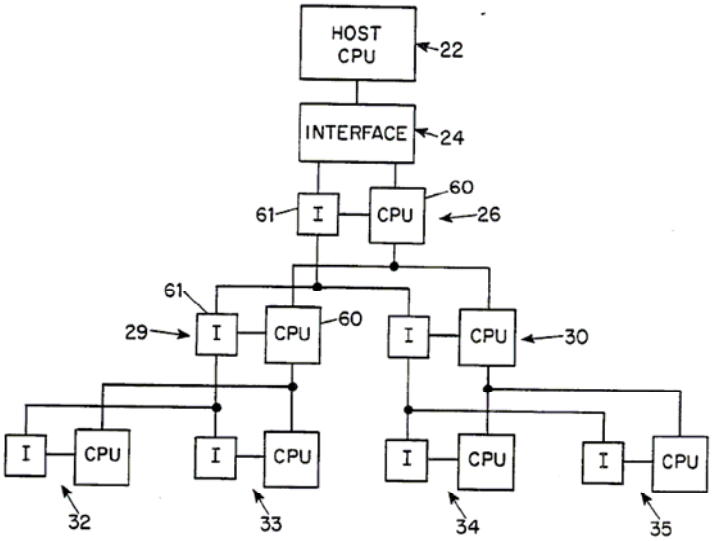

FIG. 2

Figure 2 of the '540 Patent, which the '201 Patent incorporates by reference, illustrates the claimed binary tree system and the connections between data processing elements, using the I/O device. In the Stolfo system, each pair of "I" (I/O device) and "CPU" (processor) represents a single data processing element. ('201 Patent at 4:64-66; '540 Patent at 4:16-18.) Each interface unit is connected to the interface units of a parent data processing element and two children data processing elements, except those at the root and leaves of the tree.[3] According to Stolfo, "[t]he I/O device provides interfacing between each processing element and its parent and children processing elements so as to provide significant improvements in propagation speeds through the

[3] Figure 2 also demonstrates that each processor (CPU) is connected to parent and children processors for direct processor-to-processor communications.

binary tree." ('201 Patent at 4:66-5:2.)

To realize this speed improvement, Stolfo handed control of two key operations to the I/O device: "broadcast" and "resolve." Using the speech example, in broadcast mode, program instructions and speech data are sent down from the host computer via the I/O device to each of the data processing elements in the binary tree. (*Id*. at 10:21-24.) The I/O device is specially suited for this task because it contains dedicated circuitry that moves data from one device to its children in a single clock cycle, without processor instructions. (*Id*. at 19:35-37, 21:63-22:3.)

After receiving and storing the broadcast information, each data processing element concurrently computes the instructions (*e.g.*, determines whether the speech sample matches its portion of the speech database) and stores the result. (*Id*. at 10:24-29.) If the processing element finds a match it may store the value "one," while a "zero" may be stored if no match is found.

In the resolve phase (sometimes called "determining a priority"), each I/O device — not the processor — compares the computed values (*i.e.*, the ones and zeros) from its own attached processor and its two children to determine the best result among the processing elements (*i.e.* the maximum value). (*Id*. at 20:65-21:5.) These comparisons propagate up the tree without involving the processor until a "winning" value is found in a single processing element. (*Id*. at 22:61-64; Ex. F, '201 Prosecution History at FGC794.) By removing the processor from both the data flow operations and simple comparison computations (*e.g.*, maximum operation), and replacing it with the I/O device, information in Stolfo's system moves up and down the tree quickly without time-consuming processor instructions. (*Id*. at FGC847-48.)

C. Disputed Terms

1. binary tree

IBM's Proposed Construction	FGC's Proposed Construction
an arrangement of nodes where each node has a single parent and two children nodes, except the root node, which has no parent, and the leaf nodes, which have no children	a tree where a node has a parent node (except for the root node) and zero, one or two children nodes

a. Claim Language

The term "binary tree" appears in every claim of the '201 Patent.[4] Claim 1 — the only independent claim — explains that processing elements (sometimes called "nodes")[5] are connected in a "binary tree in which ***each processing element*** except those at the extremities of the binary tree ***is connected to one parent processing element and at least first and second child processing elements***." ('201 Patent at 69:67-70:60.) IBM's construction reflects this explicit claim requirement, by explaining that ***each*** node in the binary tree must have a parent and two children, except at the root and leaf nodes.[6] FGC's construction impermissibly conflicts with the plain language of the claims, by permitting a parent node to have only one child. *White v. Dunbar*, 119 U.S. 47, 52 (1886) ("The claim is a statutory requirement, prescribed for the very purpose of making the patentee define precisely what his invention is; and it is unjust to the public … to construe it in a manner different from the plain import of its terms.").

b. Specification and Abstract

The specification uniformly describes a binary tree as an arrangement of processing elements (or nodes) "with each processing element except those in the highest and lowest levels being in communication with a single parent processing element as well as first and second (or

---

4 Any terms recited in independent claim 1, by definition, appear in every claim of the patent.

5 *See, e.g.*, '201 Patent at 26:67-27:8.

6 Despite the open-ended claim language "*at least* first and second child processing elements," the parties agree that a node in the claimed binary tree cannot have *more* than two children.

left and right) child processing elements." ('201 Patent at Abstract.) This definition of binary tree appears in the Abstract, the Background of the Invention, the Summary of the Invention, the Detailed Description, and is illustrated in Figure 2 shown above. (*Id*. at Abstract, 1:62-65, 4:56-61, 8:47-52.) Indeed, the '201 Patent never suggests that any parent node can have only one child. *Phillips*, 415 F.3d at 1316, 1321 (the specification acts as a dictionary when it defines a term used in the claims).

2. subtree

IBM's Proposed Construction	FGC's Proposed Construction
a subset of the binary tree such that each node has a single parent and two children nodes, except the root node, which has no parent, and the leaf nodes, which have no children	a partitionable portion of a tree that is less than the entire tree created by denoting a node to be the root node of the subtree

a. Claim Language

Like "binary tree," the term "subtree" appears in every claim of the '201 Patent. Claim 1 explains that the "broadcasting" and "determining a priority" (resolve) operations can be performed on the binary tree or on a subtree. ('201 Patent at claim 1.) To perform these operations — which are designed for a binary tree — the subtree must retain the properties of the original binary tree. (*Id*. at 10:53-60.) Thus, IBM's construction specifies that the subtree is a subset of the original tree that maintains the characteristics of a binary tree, *i.e.*, one parent and two children for each node. In contrast, FGC's construction defines subtree as a portion of the tree that is "less than the entire tree," without specifying the properties of the subtree.

b. Specification

The specification explains that when dividing the claimed binary tree into subtrees, the subtrees are themselves binary and function in the same way as the original binary tree. For example, "[i]n MSIMD mode, the binary tree of PEs is partitioned into a number of ***subtrees which maintain the full functionality of the ordinal tree***." ('201 Patent at 10:53-56.) The

specification is clear that the binary tree is made up of subtrees with the same basic structure as the overall tree. (*Id*. at 1:62-65 ("[E]ach processor except those at the root and leaves of the tree has a single parent processor and two children processors.")) Because IBM's construction properly explains that a subtree is a subset of the binary tree with the same structure as the overall tree, and FGC's does not, IBM's construction of subtree should be adopted.

3. without direct control of the processors of the processing elements

IBM's Proposed Construction	FGC's Proposed Construction
[broadcasting] / [determining a priority] function is performed independently by the I/O device without receiving instructions from its associated processor.	the main processor at a node is not interrupted such that very little computational overhead is required for controlling the operation

a. Claim Language

Stolfo designed his I/O device to operate independently of the processor during broadcast and resolve operations to avoid propagation delays. The circuitry that performs the broadcast and resolve operations appears in every claim of the '201 Patent as the "means for broadcasting" and "means for determining a priority." ('201 Patent at claim 1.)[7] Consistent with Stolfo's desire to remove the processor from the two operations, the claims require that both the "means for broadcasting" and the "means for determining a priority" operate "without direct control of the processors of the processing elements." (*Id*. at claim 1.) IBM's construction is consistent with the plain meaning of the words of the claim, which state that the "broadcasting" and "determining a priority" functions are performed without instructions (control) from the processor.

In contrast, FGC's proposed construction includes a number of elements and concepts that are not found in, and are inconsistent with, the claim language. First, neither the claim nor

[7] The parties have agreed on the constructions of these two means-plus-function terms. The agreed constructions appear in Exhibit A.

the patent mentions a "*main* processor" — distinguished from another unspecified processor. Instead, each processing element has ***a*** processor. (*Id*. at 4:64-66, claim 1.) Second, FGC's construction — "very little computational overhead is required for controlling the operation" — contradicts the express claim language by suggesting that the processor ***does*** control "broadcasting" and "determining a priority." The claim, however, requires that the broadcast and resolve functions occur ***without*** direct control of the processor. *See SRI Int'l v. Matsushita Corp. of Am.*, 775 F.2d 1107, 1121 (Fed. Cir. 1985) (*en banc*) ("It is the ***claims*** that measure the invention.")

b. Specification

The specification confirms that the processor has ***no*** control over the "broadcasting" and "determining a priority" operations. The problem Stolfo set out to solve was to eliminate prior art processor control over "broadcast" and "priority determination," including information flow up and down the binary tree. (*Id*. at 3:17-33.) In Stolfo's system, the I/O device — not the processor — performs the broadcast and resolve operations. (*Id*. at 5:10-16, 5:56-59.)

c. Prosecution History

As initially drafted, the '201 Patent application claims did not contain language requiring that the I/O device perform the broadcasting and resolve operations "without direct control of the processor." But those claims were rejected as obvious in light of U.S. Patent No. 4,583,164 ("Tolle") combined with U.S. Patent No. 4,543,630. (Ex. G; Ex. H.) Like Stolfo, Tolle disclosed a binary tree-structured computer with "broadcast" functionality. (Ex. G, Tolle at 5:55-63; *see also* '201 Prosecution History at FGC564.)

In response to the PTO's rejection, applicants ***argued*** that "broadcasting" in Tolle was "***directly controlled by the processors***" whereas "broadcasting" in the '201 Patent application "proceeds ***independently of the processors*** to distribute information." ('201 Prosecution History

at FGC793.) Not persuaded, the Examiner again rejected the claims. (*Id*. at FGC829-32.)

Applicants then ***amended*** application claim 1 to cement the alleged distinction over the prior art, adding the following underlined language:

> means for broadcasting information received from a parent processing element to said child processing elements, such that common information is distributed to each processing element of the binary tree or a subtree thereof without direct control of the processors of the processing elements.
>
> and
>
> means for determining a priority among respective values of information received from said child processing elements and information received from the processor with said input/output means is associated without direct control of the processors of the processing elements

(*Id*. at FGC843-44.) The applicants argued that their system broadcasted faster because, unlike the prior art, it did not have to wait for the execution of processor instructions:

> Because 'broadcasting' in Tolle is ***directly controlled by the processors*** of the cells of the binary tree ***through the execution of a storage management algorithm*** rather than through cooperation among input/output means within each cell (i.e., 'pipelining'), the time required to move data through the binary tree in Tolle is on the order of the number of cells in the binary tree multiplied by the ***execution time of the instructions required*** for moving information from a parent cell to its left and right child cells.

(*Id*. at FGC852.) The applicants further contrasted Tolle by explaining that in the claimed invention "both the broadcast and resolve operations are carried out in accordance with applicants' invention in a pipeline fashion ***without requiring the direct intervention of the processors of the processing elements***." (*Id*. at FGC847.) The argument and then amendment during prosecution make it inescapable that "without direct control" means that the input/output devices perform the broadcast and resolve operations "independently" without instructions from associated processors. *Gillespie v. Dywidag Systems Intern., USA*, 501 F.3d 1285, 1291 (Fed.

Cir. 2007) ("The patentee is held to what he declares during the prosecution of his patent."). FGC cannot now recapture systems where the processor does provide instructions to control the broadcast and resolve function.

4. each in a time on the order of the logarithm of the number of processing elements in said binary tree or subtree multiplied by the time for the broadcasting of information from a parent processing element to child processing elements connected thereto, and the time required to determine priority among values of information received from the processor of a processing element and the child processing elements connected thereto, respectively

IBM's Proposed Construction	FGC's Proposed Construction
the broadcasting operation is performed in one clock cycle, multiplied by the base 2 logarithm of the number of processing elements in the binary tree; the priority is determined in two clock cycles, multiplied by the base 2 logarithm of the number of processing elements in the binary tree	each operation is performed in a time based on the base 2 logarithm ($\log_2$) of the total number of nodes in a tree or subtree multiplied by the time required to broadcast from a parent to a child node and the time required to determine the result data between a parent and its child nodes

a. Claim Language

The claims of the '201 Patent not only identify the circuitry used to perform the "broadcasting" and "determining a priority," functions but also specify the amount of time it takes for the circuitry to perform each function. ('201 Patent at claim 1.) Thus, when the claim term begins with the phrase "each in a time," it is referring to the time it takes to perform each of the "broadcasting" and "determining a priority" functions.

The time for each operation is set forth in a mathematical formula that takes into account the capabilities of the I/O device and the structure of the binary tree. According to the claim, the time for the broadcast operation is calculated by multiplying the logarithm of the number of processing elements in the tree by the time it takes to broadcast from one parent to its child processing elements. In a binary tree, where all of the processing elements have two children,

such as is described and claimed in the '201 Patent, the logarithm[8] of the number of processing elements is another way of identifying the number of levels in the tree. Thus, the time it takes to broadcast throughout the entire tree is the number of levels in the tree multiplied by the time it takes for the broadcast to travel between levels. Similarly, the time to determine a priority throughout the tree is the number of levels in the tree multiplied by the time it takes to determine the priority and report the result at each level.

time required for the broadcast or priority determination operation at each level	X	number of levels in the binary tree	=	total time for the broadcast or priority determination operation

IBM's construction thus sets forth separate time formulas for each operation.

FGC's proposal misconstrues the claim and should be rejected because it suggests that "each operation" occurs in a time that is based on ***both*** the broadcast and priority determination functions.

b. Specification

Stolfo explains that his system improves on the prior art broadcast and priority determination functions because it uses a specialized I/O device that operates independently of the processor. ('201 Patent at 3:17-33, 4:66-6:9.) The parties agree that ***claimed*** structures used to perform these operations are depicted in Figures 7 (broadcast) and 9 (priority determination). (*See* Ex. A at 1.) Referring to Figure 7, the specification explains that it takes ***one clock cycle*** to broadcast information from parent to child. (*Id.* at 19:35-37.) Similarly, referring to Figure 9, the specification explains that it takes ***two clock cycles*** to perform the priority determination (resolve) function — one clock cycle for the comparison and one clock cycle to propagate the result to the next level in the tree. (*Id.* at 21:63-22:3.) IBM's construction reflects Stolfo's

[8] The parties agree that in the case of a binary tree, this is a "base 2 logarithm."

claimed improvement over the prior art by identifying the time it takes for the claimed structures to perform the broadcast and priority determination functions at each level. That time is then multiplied by the number of levels in the binary tree system (base 2 logarithm of the processors).

FGC ignores the claimed propagation-speed improvement by providing a tautological claim interpretation that ignores the claimed structure and requires that the broadcast and priority determinations at each level take place in ***some amount of time,*** regardless of whether that length of time matches or exceeds the allegedly unacceptable delays in the prior art. *Ballard Medical Products v. Allegiance Healthcare Corp.*, 268 F.3d 1352, 1359 (Fed. Cir. 2001) ("An inventor may use the specification and prosecution history to define what his invention is and what it is not - particularly when distinguishing the invention over the prior art.").

5. single instruction multiple data mode (dependent claim 8)

IBM's Proposed Construction	FGC's Proposed Construction
where each processing element is first loaded with its own data and then a single stream of instructions is broadcast to every processing element in the binary tree	each processing element has its own data but executes the same instruction as other processing elements.

multiple instruction multiple data mode (dependent claim 8)

IBM's Proposed Construction	FGC's Proposed Construction
where each processing element is first broadcast its local program and data and then each processing element is logically disconnected from its neighbor processing element and executes independently	each processing element has its own program and data and executes independently of its neighbor processing elements.

The '201 Patent specification explains that Stolfo's binary tree computer system can be run in one of two modes: "SIMD" and "MIMD," which refer to "single instruction stream multiple data stream" and "multiple instruction multiple data mode," respectively. (*Id*. at 10:35-38.) These modes of operation describe different ways of handling "data and/or instructions …

broadcast" throughout the tree (*Id*. at 10:33-35) and are defined in the specification:

- "[I]n SIMD mode, each PE is first loaded with its own data and then a single stream of instructions is broadcast to all PEs." (*Id*. at 10:41-43.)
- "In MIMD mode, each PE is first broadcast its local program and data and then each PE is logically disconnected from its neighbor PEs and executes independently." (*Id*. at 10:48-51.)

IBM's proposals are correct because they are identical to the definitions explicitly set forth in the specification. *Phillips*, 415 F.3d at 1321.

IV. U.S. Patent No. 6,000,024

Like the '201 Patent, the '024 Patent describes and claims a binary tree computer system. ('024 Patent at Abstract, 1:3-5.) In fact, the '024 Patent contends that it improves on the Stolfo Patents. (*Id*. at 1:40-46.) According to the '024 Patent, in prior computer systems, such as Stolfo's, "processing elements spend a substantial portion [of] their time dealing with communications up and down the tree, which reduces the processing throughput of the system."[9] ('024 Patent at 1:36-38.) The '024 Patent sought to speed the system by removing communications responsibilities from the processing elements and instead using a binary tree of bus controllers to connect the processing elements to each other and to a host computer.

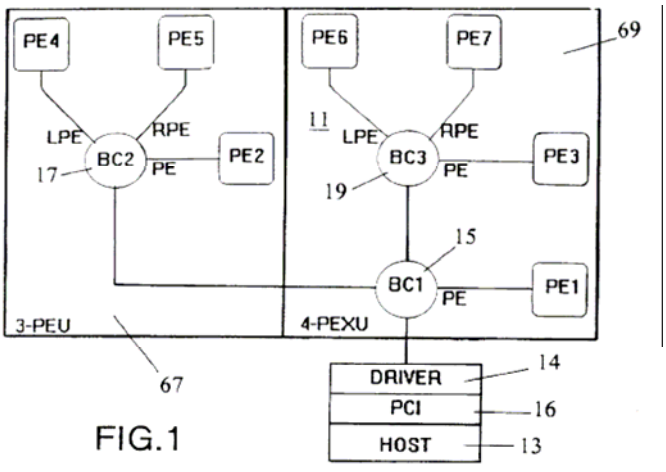

FIG.1

[9] The processing elements in the '201 Patent include the specialized I/O device. (Ex. A at 1.)

As demonstrated in Figure 1 of the '024 Patent, much like the binary tree of processing elements in the '201 Patent, the binary tree of bus controllers ("BC") in the '024 Patent is arranged so that "each bus controller except those at the extremes of the tree [is] connected to left and right child bus controllers." (*Id*. at Abstract.) Thus, in the above figure, BC1 has two children: BC2 and BC3. BC1 is referred to as a "root" bus controller. (*Id*. at 2:50-53.) The root bus controller "connects the binary tree to the host computer." (*Id*. at Abstract.) The host computer, in turn, controls the operation of the binary tree. (*Id*. at 2:64-3:1, 3:35-38.)

As shown above, "[e]ach of the bus controllers has an associated processing element attached thereto and two [additional] processing elements are connected to each of the bus controllers at the extremes of the binary tree." (*Id*. at Abstract.) Thus, root bus controller BC1 has one attached processing element, while BC2 and BC3, which are at the extremes of the tree, each have three attached processing elements. (*Id*. at 2:50-56.)

The bus controllers direct the flow of information throughout the binary tree by transferring instructions generated by the host computer for controlling the system and data between the host computer and the processing elements and between processing elements. (*See id*. at 2:64-3:1, 3:36-38.) By arranging the bus controllers in a binary tree configuration, the '024 Patent moves the binary tree communications away from the processor and to the bus controllers.

A. Disputed Terms

1. binary tree computer system

IBM's Proposed Construction	FGC's Proposed Construction
a computer system of nodes connected in a binary tree configuration	a computing system (or a partitionable portion of a computing system) with nodes connected in a binary tree configuration

binary tree configuration

IBM's Proposed Construction	FGC's Proposed Construction
an arrangement of nodes where each node has a single parent and two children nodes, except the root node, which has no parent, and the leaf nodes, which have no children	a tree arrangement where a node has a parent (except for a root node) and zero, one, or two children nodes

a. Claim Language

The terms "binary tree computer system" and "binary tree configuration" appear in each independent claim (claims 1 and 7) of the '024 Patent. Claims 1 and 7 first explain that they are directed to a "binary tree computer system" and then list the elements of that computer system, including "bus controllers connected in a binary tree configuration." Thus, the plain language of the claims demonstrates that the binary tree computer system is a computer system of nodes[10] connected in a binary tree configuration.

FGC agrees with IBM's construction, but seeks to expand the binary tree computer system to include "a partitionable portion of a computing system." The claim, however, is not directed to a mere "portion" of a computer system — it is explicitly directed to the system itself. *McCarty v. Lehigh Valley R. Co.*, 160 U.S. 110, 116 (1895) ("[W]e know of no principle of law which would authorize us to read into a claim an element which is not present.…").

As in the '201 Patent, claims 1 and 7 require that "each bus controller, except those at the extremes of the tree, are connected to left and right child bus controllers." Thus, each bus controller must have two child bus controllers, except at the extremes of the tree, where there are no children.[11] IBM's construction reflects this explicit claim requirement. FGC's construction contradicts the claim, by permitting a parent node with only one child.

---

[10] The bus controllers are sometimes referred to as "nodes." (*See, e.g.*, '024 Patent at 2:50-56.)

[11] The parties agree that bus controllers at the extremes (*i.e.*, leaves) of the tree have no child bus controllers. (*See* Ex. A at 3.)

b. Specification

The ’024 Patent specification confirms that each parent node must have two children. For example, the Abstract defines a “binary tree configuration” as an arrangement “in which each bus controller except those at the extremes of the tree [is] connected to left and right child bus controllers.” (’024 Patent at Abstract.) The same definition appears in the Summary of the Invention. (*Id*. at 1:49-51.) Indeed, the ’024 Patent consistently defines a binary tree configuration as a type of parallel computing system in which each bus controller, except at the leaves, has two children. (*Id*. at 1:8-22, 2:50-56.) Where the patentee has explicitly defined a term the explicit definition must control. *Phillips*, 415 F.3d at 1321.

The specification also says nothing about a “partitionable portion” of the binary tree. FGC knew how to claim a portion of a tree, as Stolfo did with the term “subtree” in the ’201 Patent claims. In the ’024 Patent, however, the claims never mention subtrees or partitioning.

2. host computer

IBM’s Proposed Construction	FGC’s Proposed Construction
a computer that is connected to and controls the binary tree of bus controllers	a computer connected to a network that provides access to that network.

a. Claim Language

Both independent claims 1 and 7 explain that the binary tree computer system is “for connection to and control by a host computer.” The claims, therefore, require that the “host computer” is connected to and controls the binary tree of bus controllers, as specified in IBM’s construction. Independent claim 7 and dependent claim 3 both state that the bus controllers each have “means for interpreting instructions received from the host computer,” providing further support that the host computer controls the binary tree of bus controllers.

FGC’s proposal, which adds the concept of a network and access to that network, finds no support in the claims, which say nothing about a “network.”

b. Specification

The specification confirms that the host computer is connected to and controls the binary tree of bus controllers, and does not merely connect to and provide access to an unspecified network. For example, in Figure 1 — which "illustrates the basic structure of the binary tree parallel computer system of the invention" ('024 Patent at 2:50-51) — the host computer 13 is connected to the binary tree of bus controllers through root bus controller BC1.

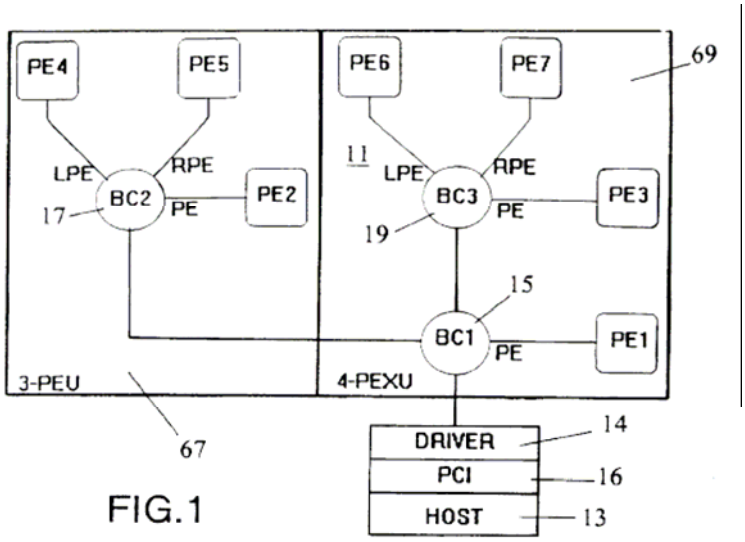

FIG.1

The specification of the '024 Patent states that "the host computer [] ***generates instructions referred to as Function Calls to control the operation of the system***." (*Id.* at 3:35-38.) The specification continues by explaining that the host computer's instructions are transmitted throughout the connected binary tree by the bus controllers, and then data from processing elements eventually is transmitted back to the host computer after the instructions are executed. (*Id.* at 2:64-3:1.)

In contrast, the '024 Patent never mentions a network (other than the binary tree of bus controllers) or a host computer providing access to it. A claim construction proposal like FGC's, which finds "no support in the intrinsic record," cannot be the correct construction. *See, e.g.*, *Old Town Canoe Co. v. Confluence Holdings Corp.*, 448 F.3d 1309, 1318 (Fed. Cir. 2006).

3. bus controllers

IBM's Proposed Construction	FGC's Proposed Construction
controllers that transfer instructions and data from the host computer to the connected processing elements, and data from the connected processing elements to the host computer over a bus	A buffer interface connecting the processing element to the bus controller and the means for writing information into the memory of the connected processing element without control of the main processor of the connected processing element

a. Claim Language

Independent claims 1 and 7 explain that the bus controllers are connected in a binary tree configuration, with attached processing elements. The claims also require that a specific bus controller — the root bus controller — be connected to the host computer. Accordingly, a "bus controller" is a controller that controls the flow of information over the busses connecting the components of the binary tree computer system.

Claims 1 and 7 further inform the proper construction of the "bus controller," specifying that each bus controller has:

- in claim 1, a "buffered interface connecting said processing element to said bus controller for transmitting instructions and data between the bus controller and the connected processing element"; and
- in claim 7, "means for interpreting instructions received from the host computer … and for passing instructions to bus controllers down the tree."

The claims therefore demonstrate that the "bus controllers" transfer instructions and data from the host computer to the processing elements, and data from the processing elements to the host computer over a bus, as in IBM's construction. *See Phillips*, 415 F.3d at 1314 ("[T]he usage of a term in one claim can often illuminate the meaning of the same term in other claims.").

FGC's proposed construction confuses the issue. First, FGC's definition of "bus controller" circularly uses the term "bus controller," thus providing no guidance to the meaning of the term. *See, e.g., O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co., Ltd.*, 521 F.3d 1351,

1362 (Fed. Cir. 2008) (explaining that the purpose of claim construction is "to clarify and when necessary to explain what the patentee covered by the claims"). Second, FGC's construction indicates that the bus controller is an interface for connecting the "means for writing." But claim 1 specifies that the bus controller itself ***includes*** the "interface" and "means for writing," not that the bus controller ***is*** the "interface." FGC's construction thus jumbles the various elements of claim 1 into a circular, meaningless, and confusing construction.

b. Specification

Consistent with the claimed invention, the specification describes the "bus controllers" as transferring instructions and data from the host computer to the processing elements, and data from the processing elements to the host computer:

> The BCxs [bus controllers] act as buffered repeaters that transfer Function Calls and data from the Host Computer to the selected PE(s), and data with its Fault Message from the selected PE to the Host Computer.

('024 Patent at 2:64-3:1.) Further confirming IBM's proposed construction, the Abstract states that bus controllers "transmit[] instructions and data between the bus controller and the processing element." (*Id*. at Abstract.) The specification also provides that the "Bus Control Nodes (BCx) operate as repeaters to act collectively as a bucket brigade to transfer data between the HIF [host interface] and the PEs, and PEs to PEs." (*Id*. at 5:38-40.) Thus, the specification confirms, as IBM proposes, that bus controllers are controllers that transfer instructions and data from the host computer to the connected processing elements, and data from the connected processing elements to the host computer over a bus.

4. root bus controller

IBM's Proposed Construction	FGC's Proposed Construction
one bus controller at the highest order position of the binary tree computer system that connects the binary tree to the host computer and which has no parent bus controller	a bus controller that is designated the highest level bus controller in the tree or subtree

a. Claim Language

Independent claims 1 and 7 describe the root bus controller as the link between the binary tree of bus controllers and the host computer. As such, the root bus controller occupies the highest order position of the binary tree computer system, as both parties agree. The claims further require that "***one*** of said bus controllers being a root bus controller … connect[s] said binary tree connected bus controllers to said host computer." Finally, the parties agree that by virtue of its place in the tree, the root bus controller has no parent bus controller. Indeed, in its proposed construction for binary tree configuration (above), FGC explains that every node, ***except the root node***, has a parent. IBM's proposal reflects these characteristics of the root bus controller that flow from the claims.

FGC's construction impermissibly contradicts the claim language — which explicitly requires a connection to the host computer by "one" root bus controller. FGC suggests that any bus controller "designated" as the highest level is the root bus controller instead of the one bus controller that connects the binary tree of bus controllers to the host computer. But that is not what the claim requires. *White v. Dunbar*, 119 U.S. 47, 52 (1886) ("The claim is a statutory requirement, prescribed for the very purpose of making the patentee define precisely what his invention is; and it is unjust to the public … to construe it in a manner different from the plain import of its terms.").

FGC also, again, impermissibly injects the concept of a subtree, suggesting that there can be multiple root nodes for multiple subtrees. ***But the claims say nothing about subtrees***, and the claim designates "one" bus controller as the root, the one that connects the entire tree (not an undesignated subtree) to the host computer.

b. Specification

Consistent with the claim language, the specification uniformly describes the "root bus controller" as the single bus controller that connects the binary tree to the host computer:

- "***One*** of the bus controllers is a root bus controller that connects the binary tree to the host computer." ('024 Patent at Abstract.)
- "***[O]ne*** of the bus controllers being a root bus controller for connecting the tree to the host computer." (*Id*. at 1:52-53.)
- "Node BC1 15 is the root node and attaches the tree to the host 13 through a driver 14 an [sic] interface, such as PCI bus 16." (*Id*. at 2:41-42.)
- "Each node BCx is connected upstream to a parent node, ***except for the root node BC1, which is connected to the host***." (*Id*. at 2:51-53.)

Figures 1, 2, 7 and 8 illustrate root node BC1 having children BC2 and BC3 and being the single connection to the host computer. For example, in Figure 8 reproduced immediately below, the root BC1 is the one connection to the host computer, and has two children BC2 and BC3 and four grandchildren BCs 4, 5, 6 and 7.

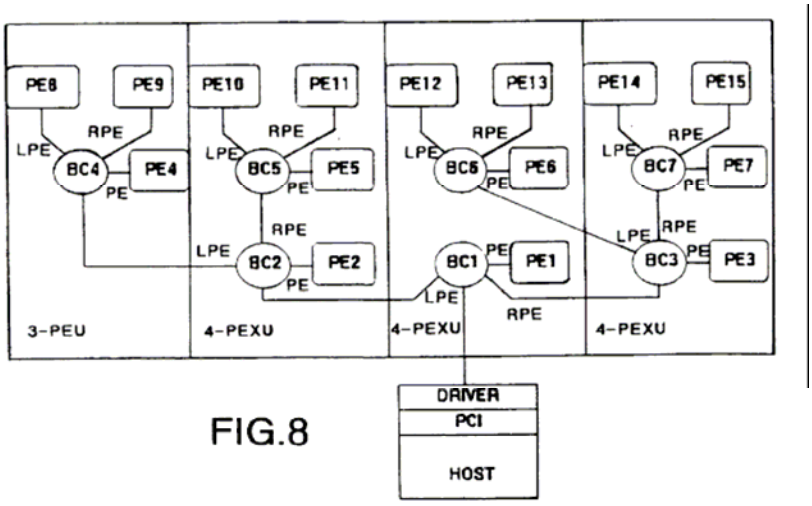

FIG.8

Conversely, the specification never discusses a subtree of bus controllers, as FGC proposes. The Court should therefore adopt IBM's proposed construction.

Dated: July 1, 2009

By: s/ Alan S. Kellman

John Desmarais
Steven Cherny
Alan S. Kellman
Karen Tu
Christopher Gerson
KIRKLAND & ELLIS
Citigroup Center
153 East 53rd Street
New York, New York 10022-4675
Telephone: (212) 446-4800
Facsimile: (212) 446-4900
Email: john.desmarais@kirkland.com
steven.cherny@kirkland.com
alan.kellman@kirkland.com
karen.tu@kirkland.com
christopher.gerson@kirkland.com

Attorneys for Defendant
INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF SERVICE

The undersigned, an attorney, hereby certifies that on July 1, 2009, a true and correct copy of the foregoing **IBM'S OPENING *MARKMAN* BRIEF** was served upon the following individuals as follows:

VIA ELECTRONIC MAIL

Gerard Norton, Esq.
Fox Rothschild LLP
Princeton Pike Corporate Center
997 Lenox Drive, Building 3
Lawrenceville, NJ 08648-2311
gnorton@foxrothschild.com

M. Michael Lewis, Esq.
Fox Rothschild LLP
100 Park Avenue
Suite 1500
New York, NY 10017
mlewis@foxrothschild.com

Bradley J. Hulbert, Esq.
McDonnell Boehnen Hulbert & Berghoff LLP
300 South Wacker Drive
Chicago, IL 60606
hulbert@mbhb.com

Daniel Adam Schnapp, Esq.
Fox Rothschild LLP
425 Park Avenue
New York, NY 10022
dschnapp@foxrothschild.com

George I. Lee, Esq.
McDonnell Boehnen Hulbert & Berghoff LLP
300 South Wacker Drive
Chicago, IL 60606
lee@mbhb.com

VIA FEDEX OVERNIGHT

Bradley J. Hulbert, Esq.
McDonnell Boehnen Hulbert & Berghoff LLP
300 South Wacker Drive
Chicago, IL 60606
hulbert@mbhb.com

George I. Lee, Esq.
McDonnell Boehnen Hulbert & Berghoff LLP
300 South Wacker Drive
Chicago, IL 60606
lee@mbhb.com

Dated: July 1, 2009

s/ Alan S. Kellman